

What is claimed is:

- FI SUB 81-82
1. A method of compressing an image having at least three textures, comprising:
 - generating a map representing boundaries
 - 5 separating regions in said image; and
 - generating pointers, each of said pointers associating one of said regions with one of said textures.
 2. The method of claim 1 wherein said map
10 comprises a bitmap, said boundaries comprise pixels of a first value, and said regions comprise pixels of values other than said first value.
 3. The method of claim 1 further comprising assigning codes to said textures in said image.
 - 15 4. The method of claim 3 wherein each of said pointers includes one of said codes.
 5. The method of claim 1 wherein each of said pointers includes a location in one of said regions.
 6. The method of claim 5 wherein each of said
20 pointers comprises a single location.
 7. The method of claim 1 wherein each of said regions comprises a single one of said textures.
 8. The method of claim 1 wherein said boundaries comprise a first one of said textures.
 - 25 SUB 82 82 9. The method of claim 8 wherein generating said map comprises converting each pixel in said image which

is not said first one of said textures to a second one of said textures.

10. The method of claim 9 wherein generating said pointers comprises finding a location in each of said regions which is not said second one of said textures.

~~Sub E3~~
~~11. The method of claim 1 wherein said map comprises a bitmap having one bit per pixel.~~

~~12. The method of claim 11 further comprising encoding said map.~~

10 13. The method of claim 12 wherein said step of encoding comprises run-length-encoding.

~~Sub E4~~
~~14. A method of compressing an image having at least three textures and at least two regions, comprising:~~

~~15 assigning a code for each of said textures in said image;~~

~~generating pointers, each of said pointers associating one of said regions with one of said textures, each of said pointers comprising a location and a code; and~~

~~generating a map, representing boundary pixels of a first one of said textures separating said regions in said image, by converting each pixel in said image not of said first one of said textures to a second one of said textures.~~

15. A data structure comprising:
a map representing boundaries separating regions in an image; and

pointers, each associating a region with a texture.

16. The data structure of claim 15 further comprising a palette associating each of said textures 5 with a code.

17. The data structure of claim 15 wherein each of said pointers includes a location and a code associated with a texture.

18. The data structure of claim 17 wherein each 10 of said pointers comprises a single location and a single code.

ES JB
~~19. The data structure of claim 15 wherein said map comprises a bitmap having one bit per pixel.~~

20. The data structure of claim 19 wherein said 15 bitmap is encoded.

21. The data structure of claim 20 wherein said bitmap is run-length-encoded.

F3 Sub D3B6 A3E6
~~22. A method of decompressing an image having at least three textures, comprising:
20 providing a map representing boundaries separating regions;
referencing a pointer to determine one of said textures associated with one of said regions; and
filling said one of said regions in said map with
25 said determined one of said textures.~~

E6 JB BT
~~23. The method of claim 22 wherein said map comprises a bitmap having one bit per pixel.~~

24. The method of claim 23 further comprising decoding said bitmap.

25. The method of claim 24 wherein said decoding comprises run-length-decoding.

5 26. The method of claim 23 further comprising converting said bitmap from one bit per pixel to multiple bits per pixel.

Sub 10
B8 E7
27. The method of claim 22 wherein filling said one of said regions comprises
referencing a pointer to determine a location, and
converting one of said regions containing said determined location into said determined one of said textures.

15 28. The method of claim 27 wherein filling said one of said regions further comprises
determining a function associated with said determined one of said textures,
converting, according to said function, each pixel in said region containing said determined location into a
20 pixel color.

29. The method of claim 28 wherein converting said each pixel comprises seed filling.

30. The method of claim 29 wherein said seed filling is commenced at said determined location.

Sub 25
A4 E8
31. A method of displaying an image having at least three textures, comprising:
providing a map representing boundaries separating regions;

referencing a pointer to determine one of said textures associated with one of said regions;

filling said one of said regions in said map with said determined one of said textures; and

5 overlaying said image on a background.

32. The method of claim 31 wherein said steps of extracting, filling, and overlaying are repeated for a succession of images to create the illusion of motion.

33. A method of displaying an image having at
10 least three textures, comprising:

generating a map representing boundaries separating regions in said image;

generating pointers, each of said pointers associating one of said regions with one of said
15 textures;

referencing said pointers to determine said one of said textures associated with said one of said regions;

filling said one of said regions in said map with said determined one of said textures; and

20 overlaying said image on a background.

add
A5
B11
B6
F9
F6

add
H1